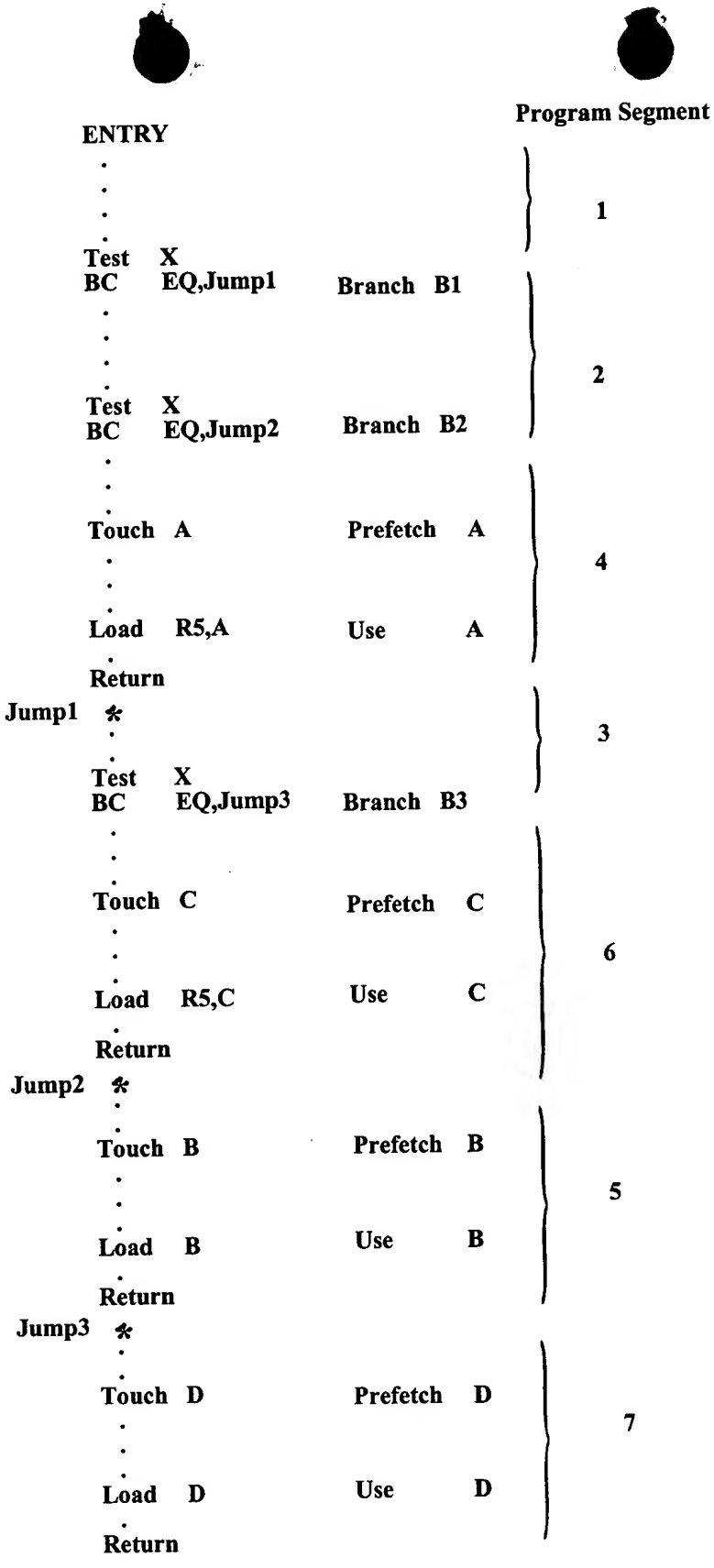

Program Segment
ENTRY
Test X
BC EQ,Jump1
Branch B1
Test X
BC EQ,Jump2
Branch B2
Touch A
Prefetch A
Load R5,A
Use A
Return
Jump1
Test X
BC EQ,Jump3
Branch B3
Touch C
Prefetch C
Load R5,C
Use C
Return
Jump2
Touch B
Prefetch B
Load B
Use B
Return
Jump3
Touch D
Prefetch D
Load D
Use D
Return
1
2
4
3
6
5
7

Figure 1

Control Flow Graph Of Program

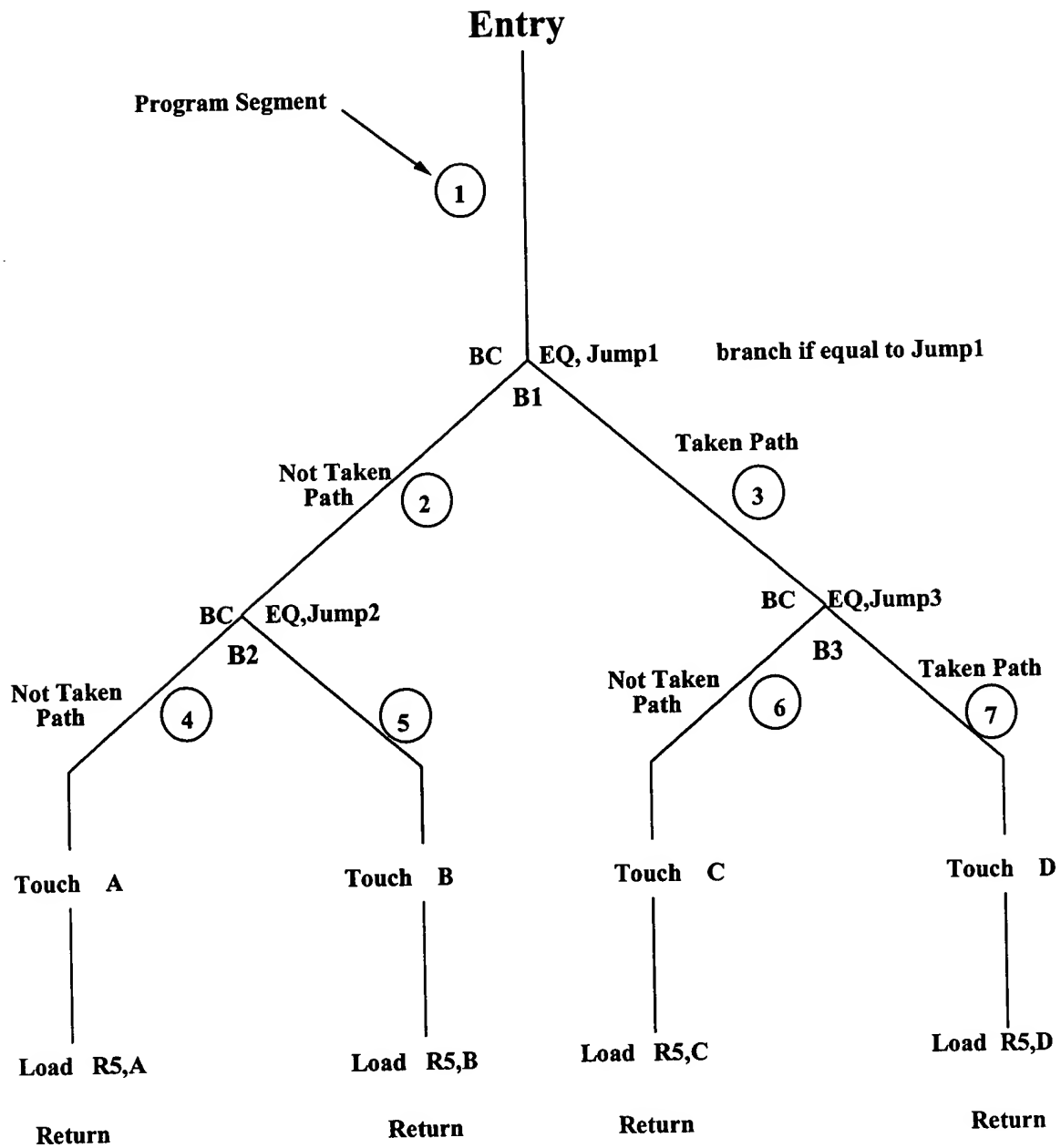


Figure 2

Control Flow Graph Of Program

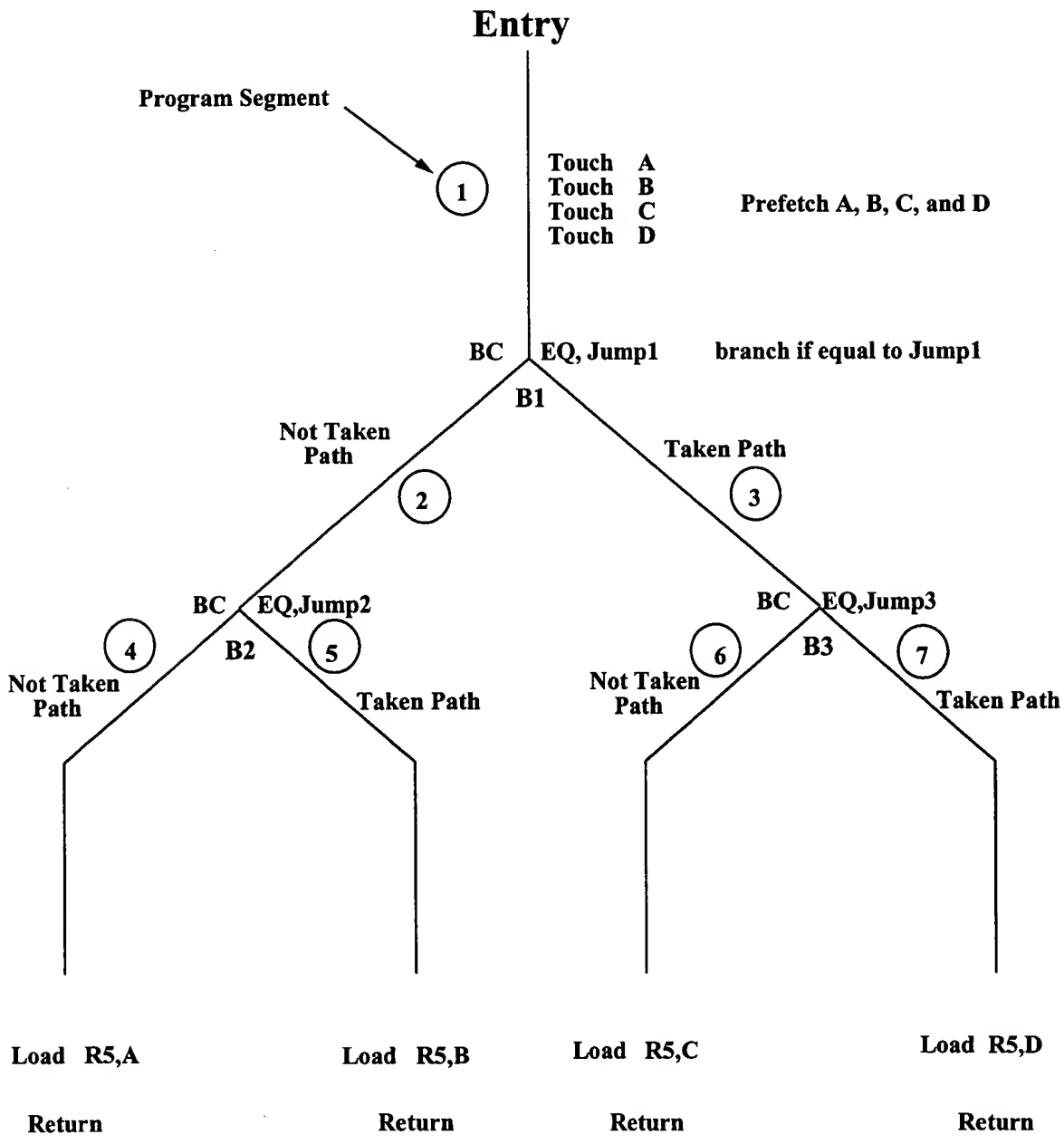
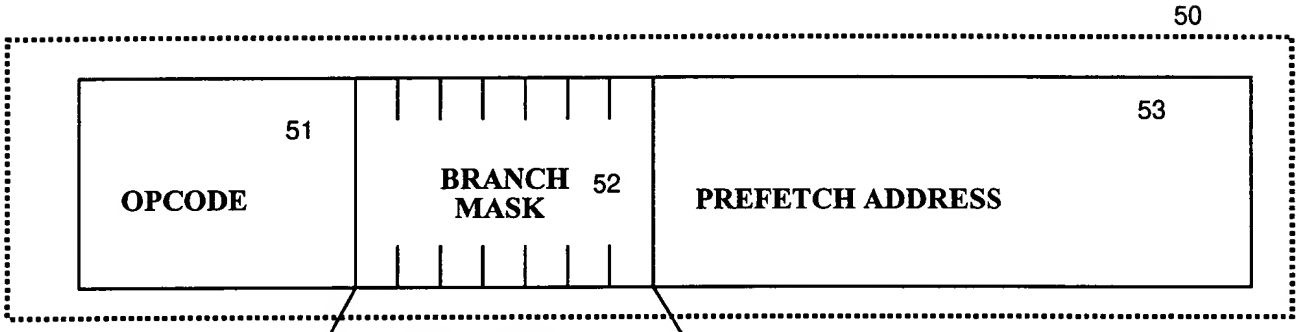


Figure 3

TOUCH INSTRUCTION FORMAT



Sub-Field 1 = Execution Action of First Branch After Touch Instruction

Sub-Field 2 = Execution Action of Second Branch After Touch Instruction

.
.
.

Sub-Field N = Execution Action of Nth Branch After Touch Instruction

The Branch Action Can Be One Of The Following Values:
T = The Branch Is Taken
N = The Branch Is Not-Taken
D = The Branch Can Either Be Taken Or Not-Taken, 'Don't Care'

Prefetch Address Can Denote A Base Register + Displacment or A Relative Offset From the Touch Instruction

Figure 4

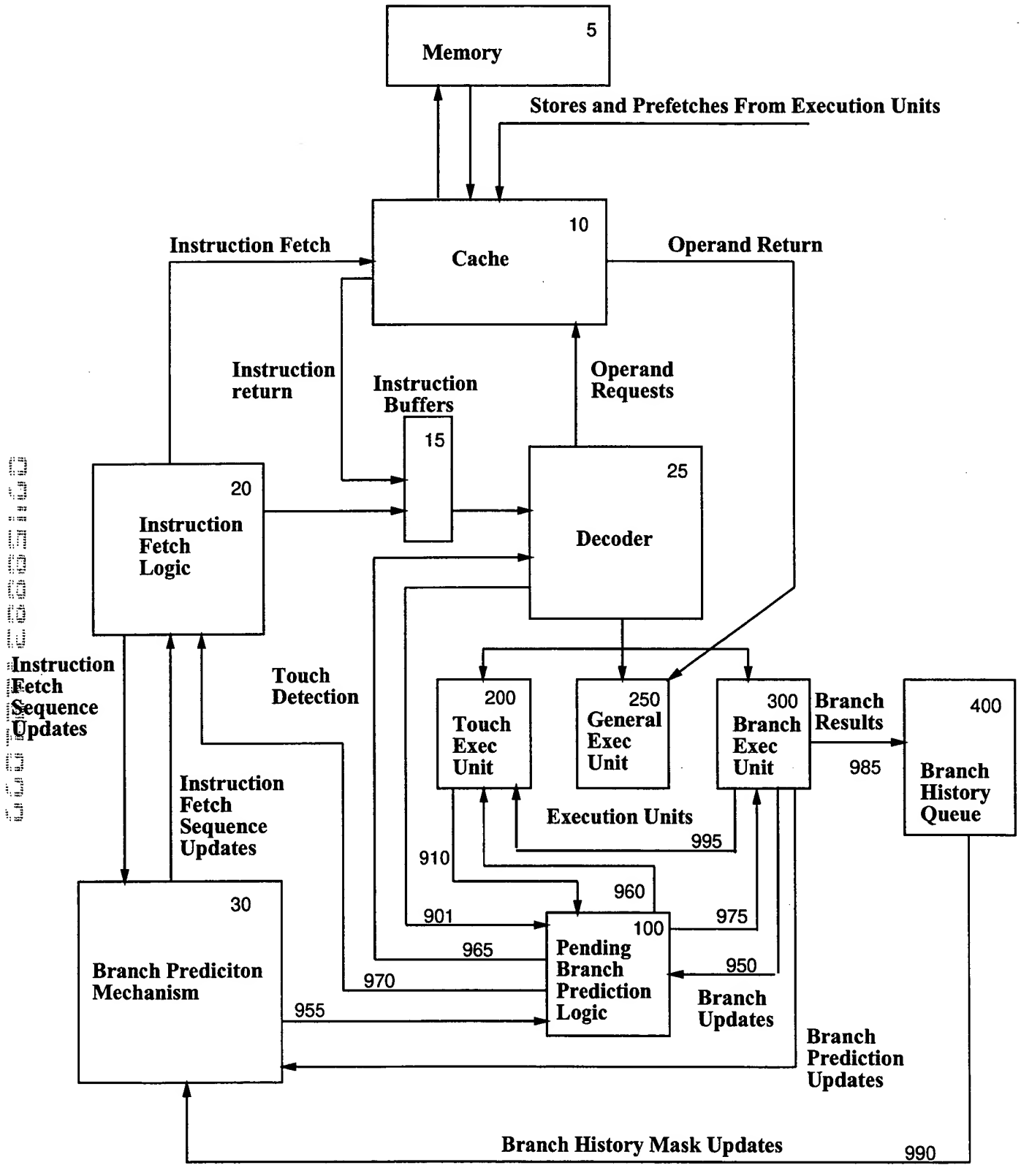

Memory
5
Stores and Prefetches From Execution Units
Cache
10
Instruction Fetch
Operand Return
Instruction return
Instruction Buffers
15
Operand Requests
Instruction Fetch Logic
20
Decoder
25
Instruction Fetch Sequence Updates
Touch Detection
200
Touch Exec Unit
250
General Exec Unit
300
Branch Exec Unit
Branch Results
985
400
Branch History Queue
Execution Units
995
910
960
901
975
100
Pending Branch Prediction Logic
965
950
970
Branch Updates
955
Branch Prediciton Mechanism
30
Instruction Fetch Sequence Updates
Branch Prediction Updates
Branch History Mask Updates
990

Figure 5

Pending Branch Prediction Logic

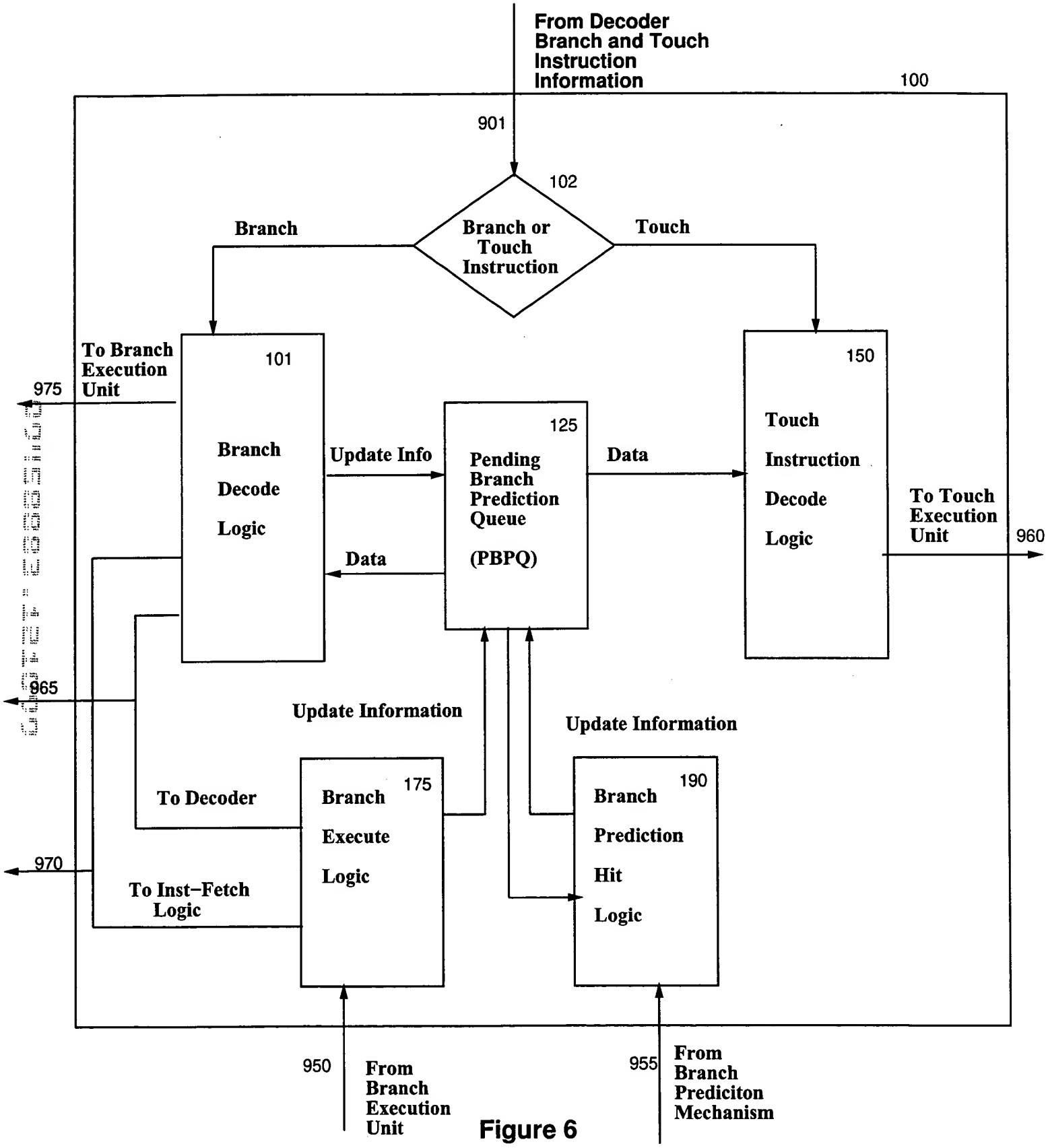


Figure 6

BRANCH HISTORY QUEUE

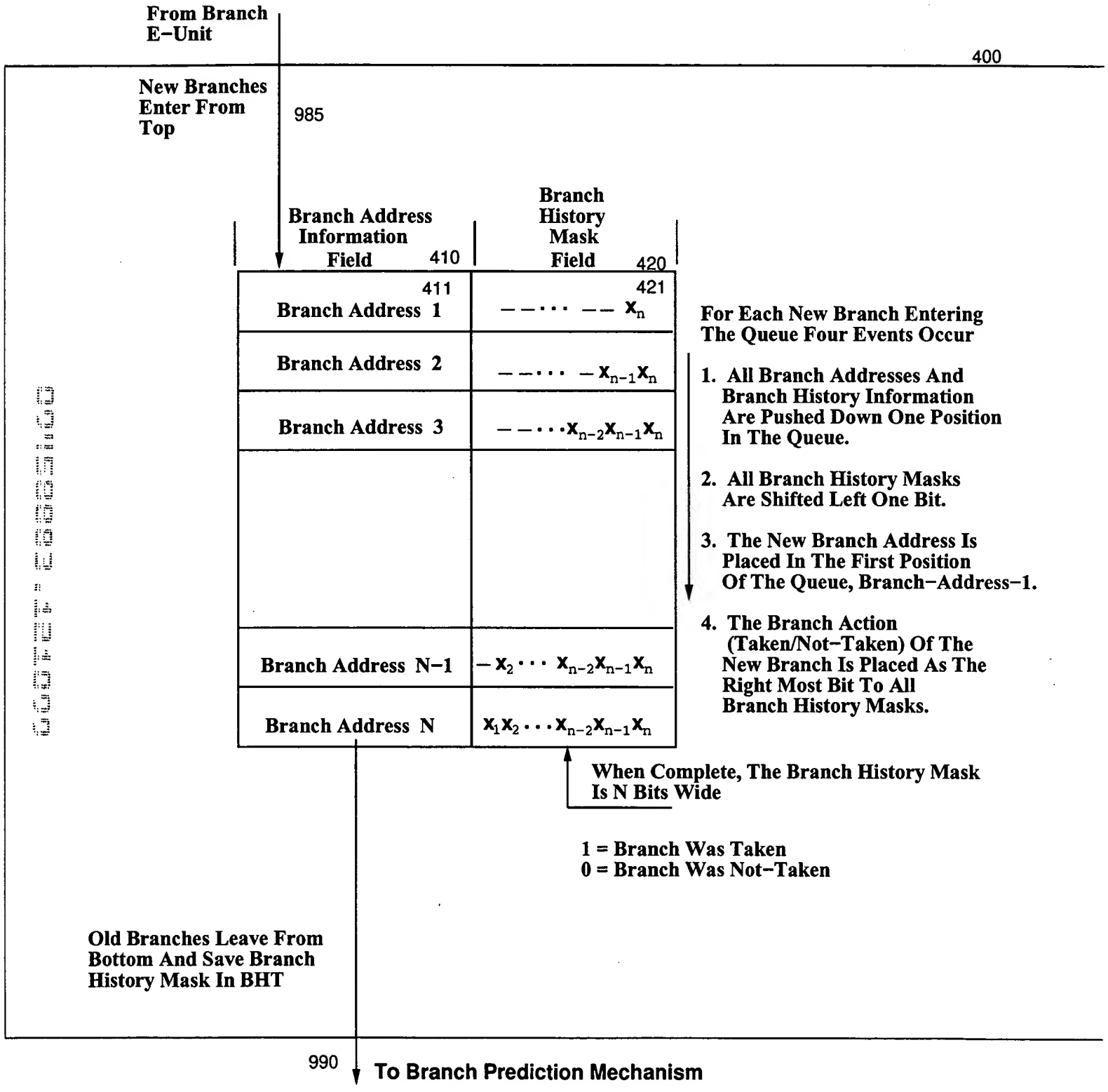
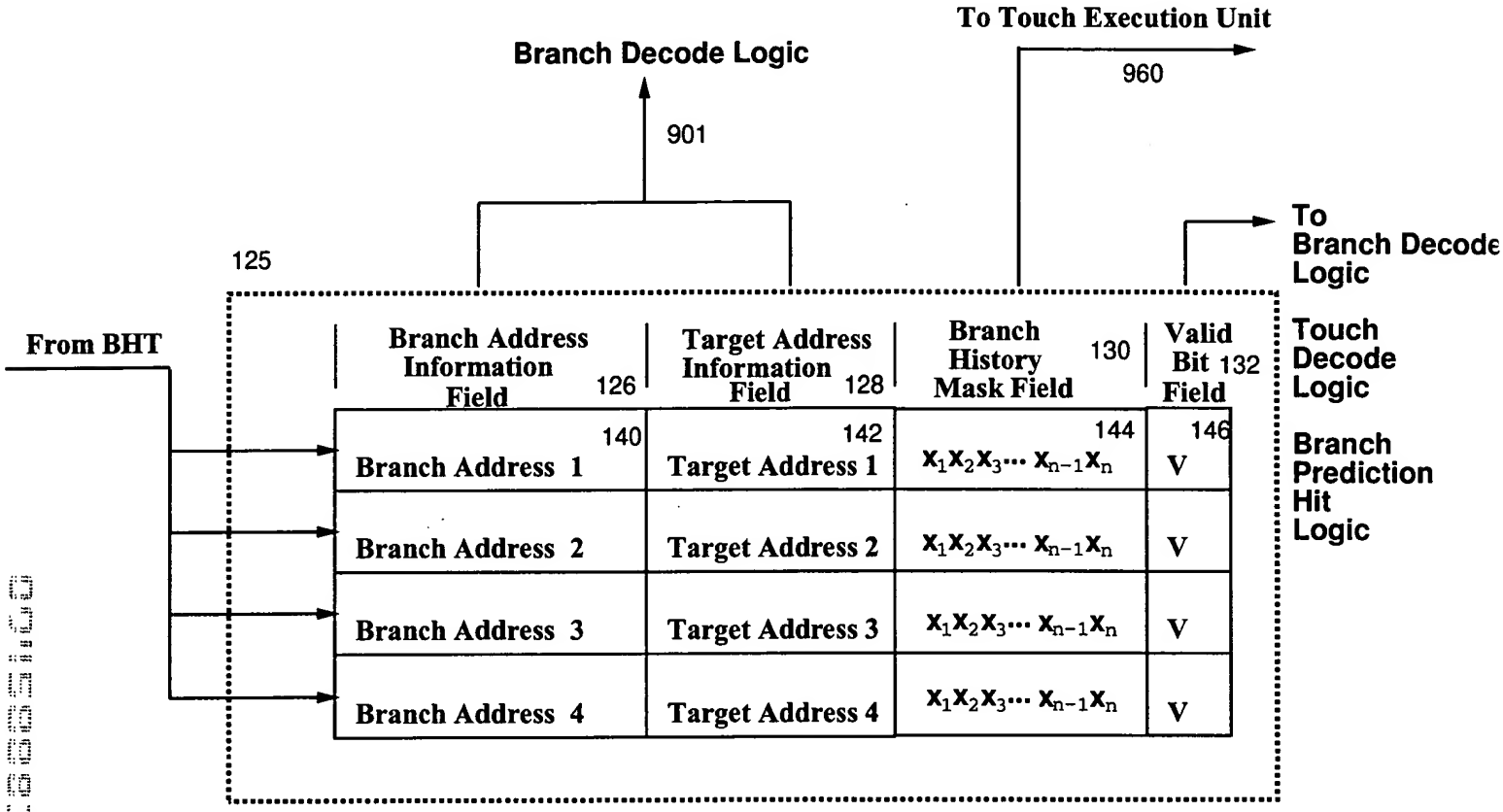


Figure 7

PENDING BRANCH PREDICTION QUEUE



Branch History Mask Records The Branch Action Of Last N Branches
1 = Taken
0 = Not-Taken

Branch Address Information

Address Of Predicted Branch Detected During Branch Prediction

Target Address Information

Predicted Target Address of Branch Detected During Branch Prediction

Valid Bit
1 = Valid
0 = Invalid

Figure 8

Branch Prediciton Mechanism Hit Logic

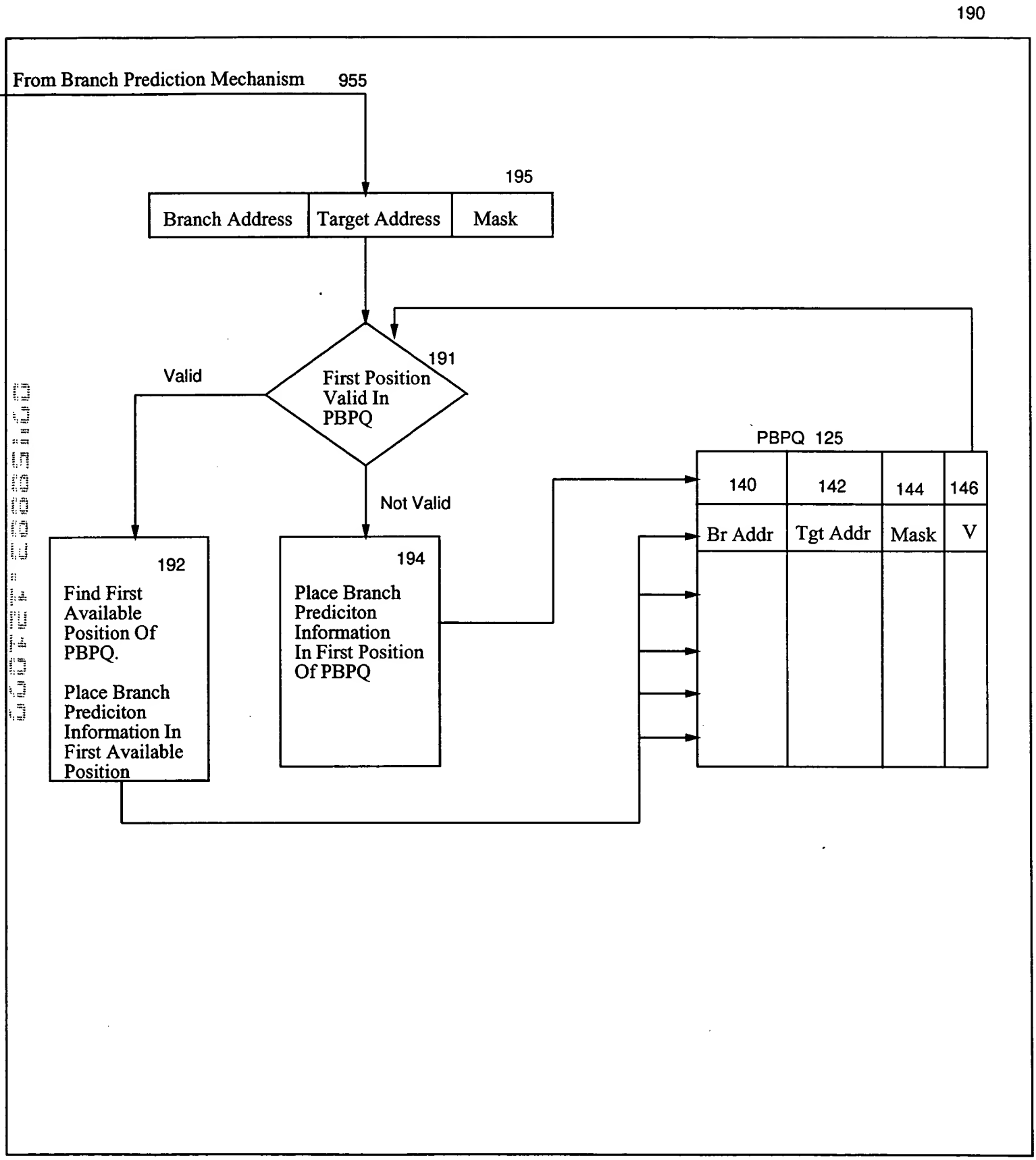


Figure 9

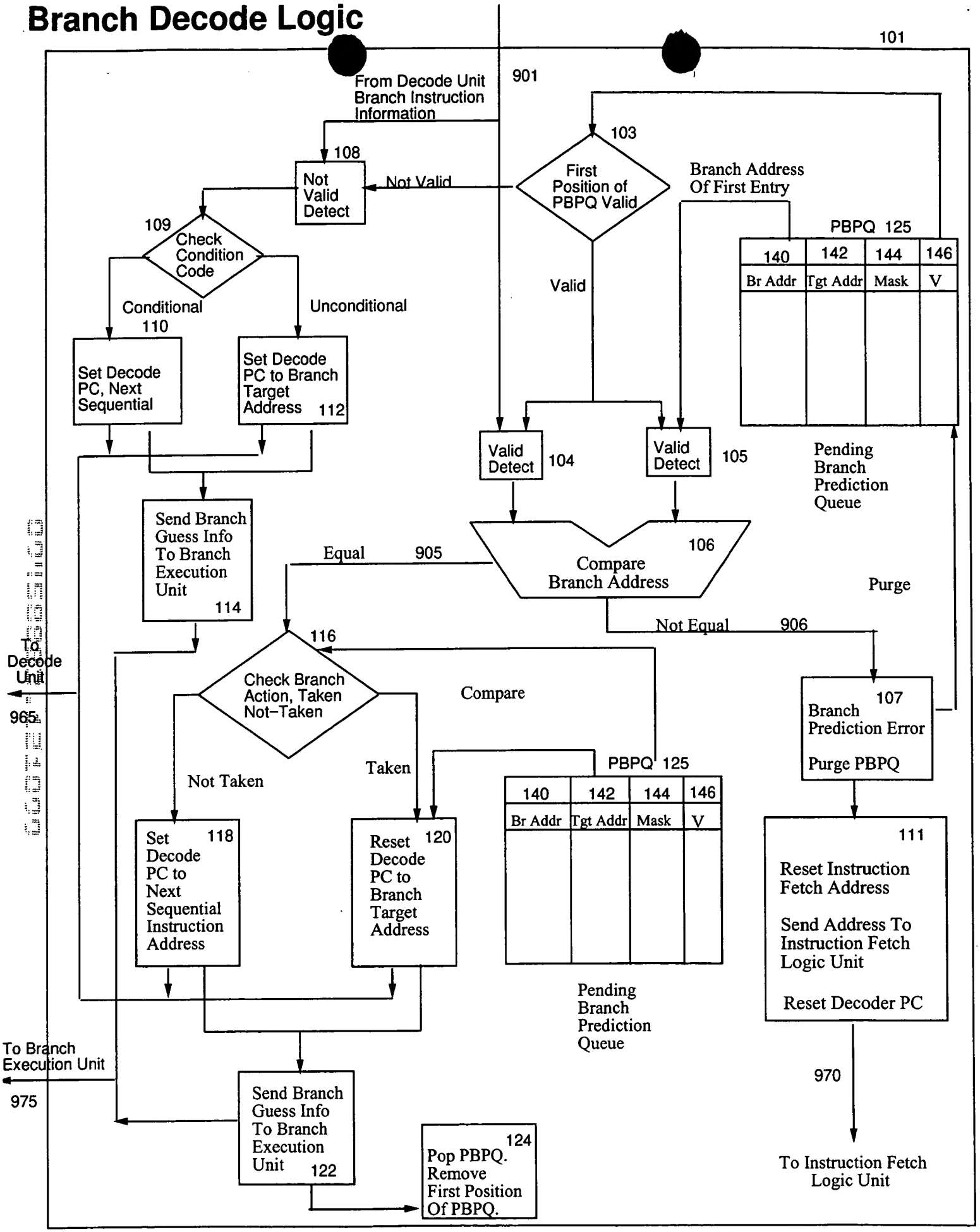

Branch Decode Logic
101
From Decode Unit Branch Instruction Information
901
103
First Position of PBPQ Valid
Branch Address Of First Entry
108
Not Valid Detect
Not Valid
109
Check Condition Code
Conditional
Unconditional
110
Set Decode PC, Next Sequential
Set Decode PC to Branch Target Address 112
PBPQ 125
140 142 144 146
Br Addr Tgt Addr Mask V
Valid
Valid Detect 104
Valid Detect 105
Pending Branch Prediction Queue
Send Branch Guess Info To Branch Execution Unit 114
106
Compare Branch Address
Equal 905
Purge
Not Equal 906
116
To Decode Unit
965
Check Branch Action, Taken Not-Taken
Compare
107
Branch Prediction Error
Purge PBPQ
Not Taken
Taken
PBPQ 125
140 142 144 146
Br Addr Tgt Addr Mask V
Set Decode PC to Next Sequential Instruction Address 118
Reset Decode PC to Branch Target Address 120
Pending Branch Prediction Queue
111
Reset Instruction Fetch Address
Send Address To Instruction Fetch Logic Unit
Reset Decoder PC
To Branch Execution Unit
975
Send Branch Guess Info To Branch Execution Unit 122
124
Pop PBPQ. Remove First Position Of PBPQ.
970
To Instruction Fetch Logic Unit

Figure 10

Touch Instruction Decode Logic

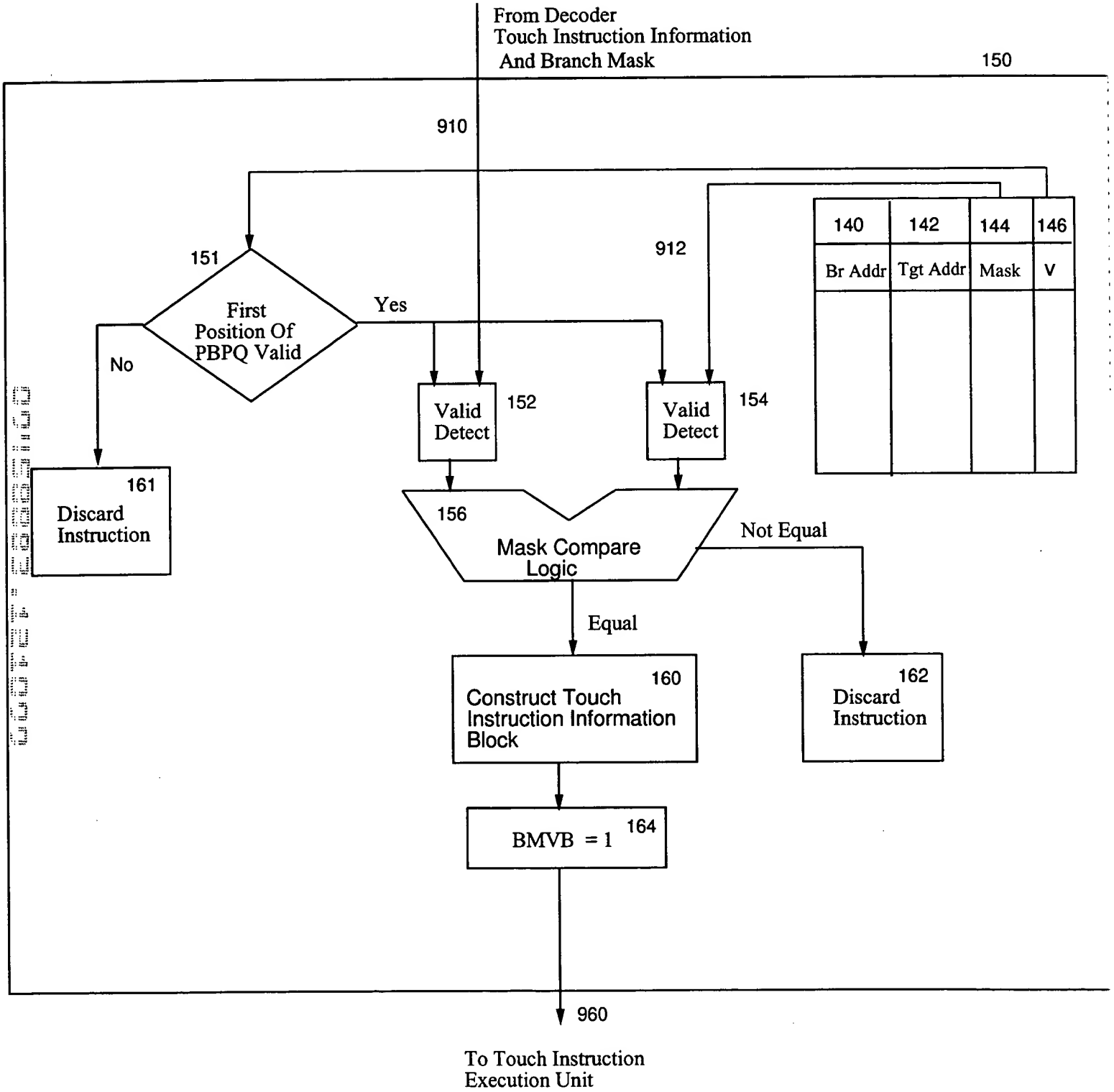


Figure 11

Branch Mask Compare Logic

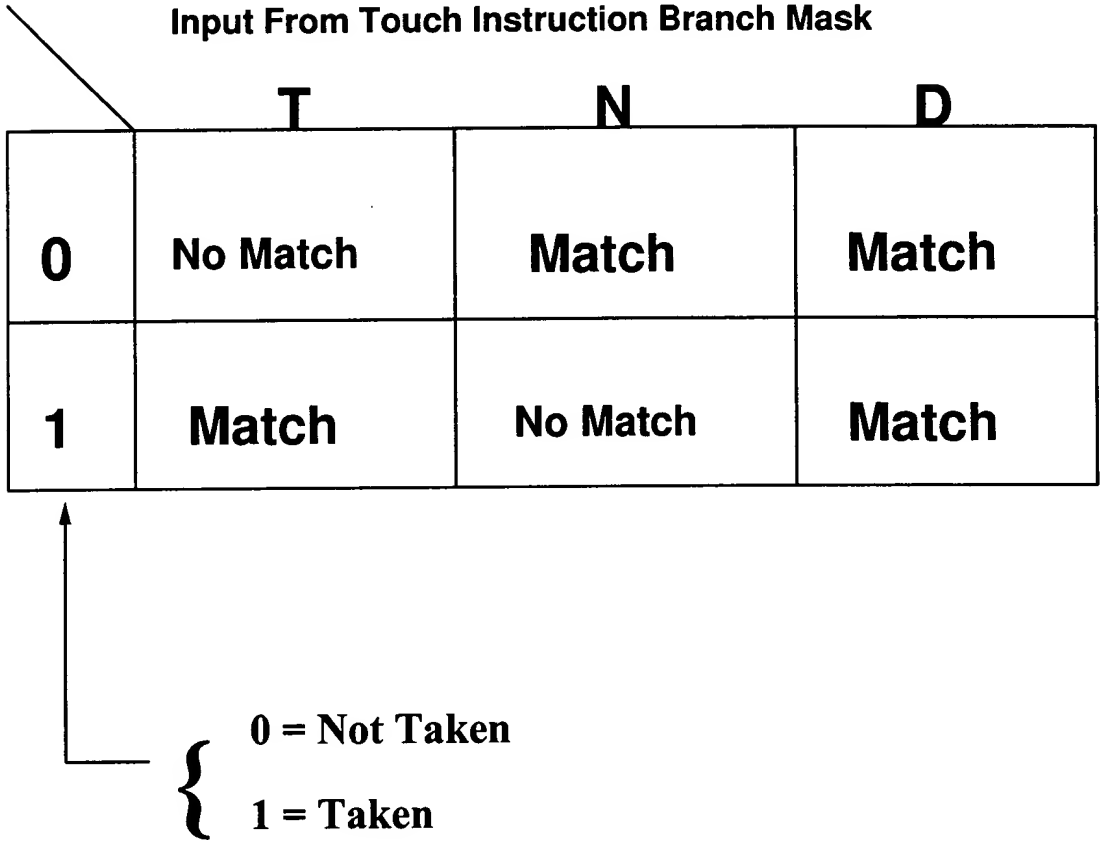
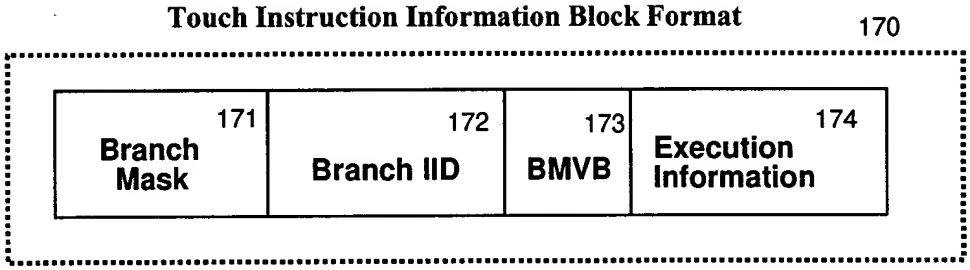


Figure 12



Branch Mask = Branch Mask Contained In Touch Instruction

Branch IID = Instruction Identifier Of Last Branch Decoded

BMVB = Branch Mask Validation Bit

Execution Information = Address Of Item To Prefetch, ...

Figure 13

Branch Execute Logic

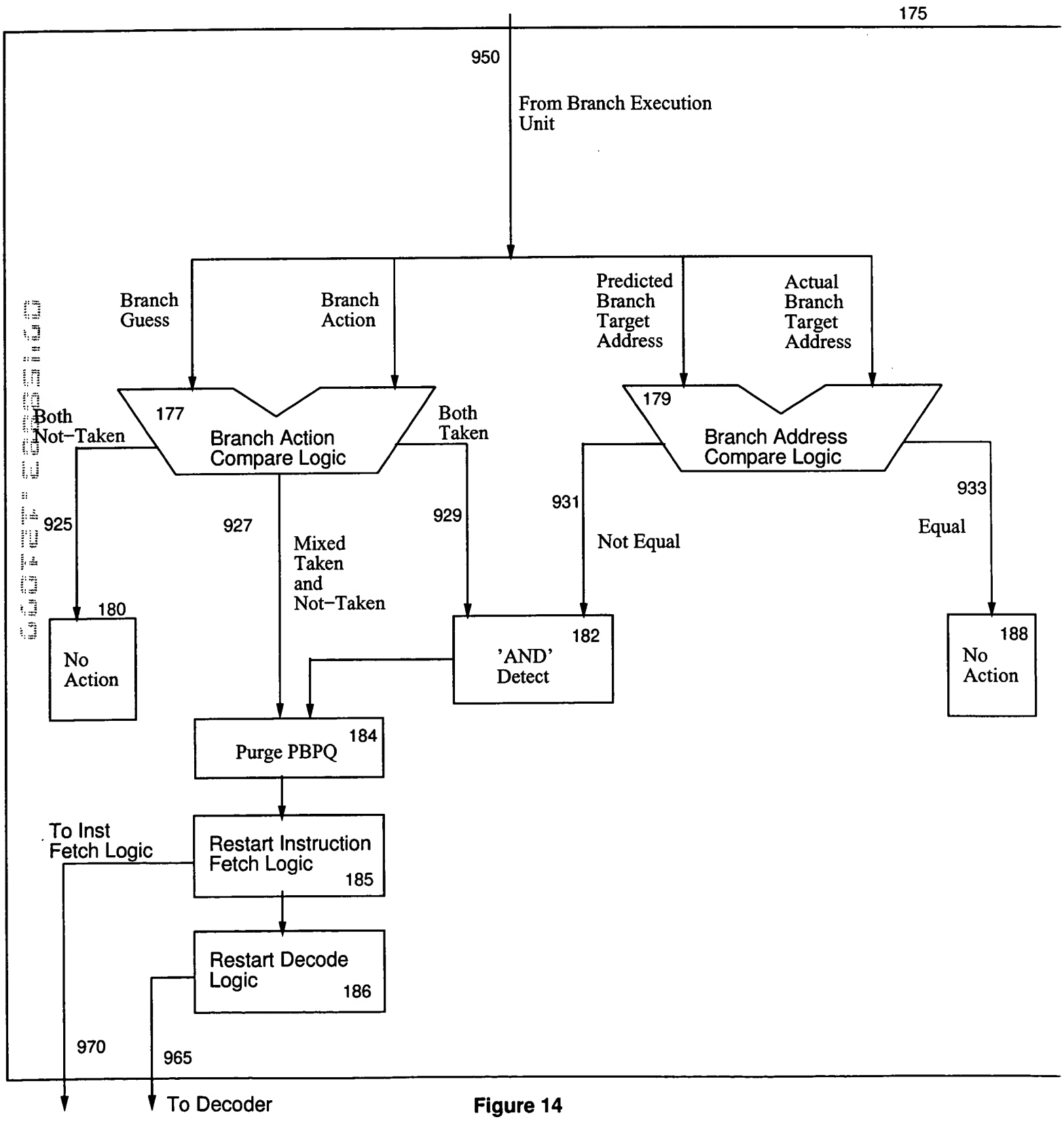


Figure 14

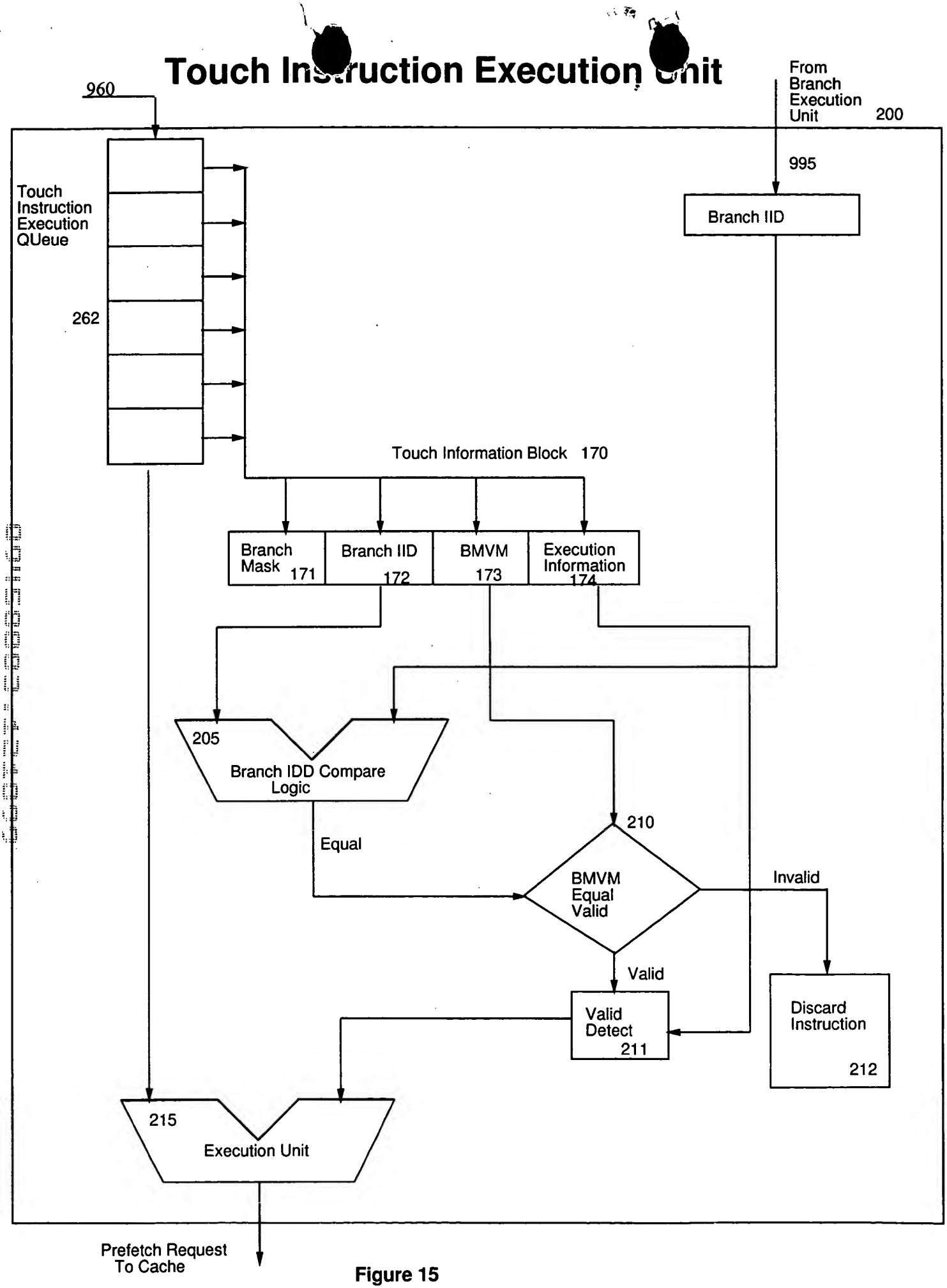

Touch Instruction Execution Unit
960
From Branch Execution Unit 200
995
Branch IID
Touch Instruction Execution QUeue
262
Touch Information Block 170
Branch Mask 171
Branch IID 172
BMVM 173
Execution Information 174
205
Branch IDD Compare Logic
Equal
210
BMVM Equal Valid
Invalid
Valid
Valid Detect 211
Discard Instruction 212
215
Execution Unit
Prefetch Request To Cache

Figure 15